Esteganografia por Espalhamento usando chaves assimétricas

Paulo Costa*
Jorge Augusto Hongo†
Anderson Rocha‡

Abstract

Foi desenvolvido um sistema de esteganografia por transposição, isto é, os bits da mensagem são armazenados de forma espalhada e aparentemente aleatória sobre o objeto de cobertura.

Em seguida, o sistema foi extendido para permitir o uso de chaves assimétricas, tanto por Diffie–Hellman quanto por Elliptic curve Diffie–Hellman. Desta forma, apenas o remetente e o destinatário são capazes de extrair a mensagem, além de ocorrer a certificação implicita do remetente (Assinatura).

Por último, a técnica é melhorada de forma a calcular uma probabilidade de uso distinta para cada bit disponível no objeto de cobertura, de forma a priorizar o armazenamento da mensagem nas regiões de detecção mais difícil.

1. Introdução

O problema de manter uma mensagem segura por meio de esteganografia e criptografia enfrenta duas dificuldades em particular: o aumento no tamanho da mensagem original quando esta é criptografada e o uso de senhas (chaves simétricas) previamente estabelecidas pelos usuários. A primeira dificuldade é conhecia por limitar o tamanho da mensagem original e aumentar o número de bits modificados, facilitando a detecção de artefatos no objeto de cobertura, enquanto a segunda é conhecida por ser vulnerável a diversas formas de ataque, como Plain-Text Attack e Differential Cryptanalysis.

Há também uma preocupação dentro do campo da criptografia em preservar o tamanho do arquivo a ser criptografado, conhecido como format-preserving encryption (FPE), com exemplos na segurança de números de cartões de crédito e na preservação de arquivos jpeg. Uma vez que essas mensagens são tanto criptografadas e ocultas via esteganografia quando precisam ser mantidas em segurança, o programa em questão pretende oferecer uma alternativa nos casos nos quais esse tipo de informação costuma ser tratado com ambas sem precisar fazer uso direto da criptografia.

*Is with the Institute of Computing, University of Campinas (Unicamp). **Contact**: eu@paulo.costa.nom.br

†Is with the Institute of Computing, University of Campinas (Unicamp). **Contact**: jorgeahongo@gmail.com

‡Is with the Institute of Computing, University of Campinas (Unicamp). **Contact**: anderson.rocha@ic.unicamp.br

2. Estado-da-Arte

Esforços direcionados no campo da esteganografia envolvendo preservação do tamanho da mensagem incluem transformação discreta no wavelet e quantização de índice modularizado. Esforços envolvendo o uso de chaves assimétricas ainda focam a capacidade de manter a chave secreta segura e sem levantar suspeita. E o campo da criptografia ainda avança em garantir a geração de mensagens criptografadas sem aumento do seu tamanho para todos os tipos e mídia.

Dentro do nosso conhecimento, nenhuma pesquisa se encontra direcionada em utilizar uma chave assimétrica previamente conhecida para obter uma forma mais segura de transmissão de dados que seja capaz de, simultaneamente, esconder, conservar e proteger uma mensagem de tentativas de decifrá-la.

3. Solução Proposta

Nossa solução é sistema de esteganografia combinado com uma cifra transposição, isto é, os bits da mensagem são armazenados de forma espalhada e aparentemente aleatória sobre o objeto de cobertura. A solução pode ser vista em 3 camadas:

- Na primeira camada, temos o objeto de cobertura, e a forma como os dados serão armazenados neste.
- Na segunda camada, temos o algoritmo de espalhamento, responsável por distribuir os bits da mensagem original sobre o objeto de cobertura de forma pseudo-aleatória.
- Na terceira camada, é gerada a entropia para o PRNG. Essencialmente, trata-se da "senha" para extrair a mensagem.

3.1. Objeto de Cobertura

Esta camada modela a forma como os dados são armazenados no objeto de cobertura. É a única responsável por definir a capacidade e robustez da técnica. Adicionalmente, é responsável pela dificuldade de detecção da mensagem, portanto divide, com as demais camadas, a responsabilidade pela segurança.

Apesar de termos nos focado em Imagens, nossa solução é capaz de trabalhar com quaisquer objetos de cobertura que se apresentem como uma coleção de bits que podem ser lidos ou alterados.

A partir desta interface simples, implementamos diversos tipos de objetos de cobertura:

- Raw - Em um arquivo com N bytes, armazena 8*N bits. Ou seja, não esconde nada, é usado apenas para testes.
- Texto - Armazena um bit a cada palavra do texto, alternando a primeira letra da palavra entre maiúscula e minuscula. A capacidade de armazenamento é pequena e é facilmente detectada por um inspeção visual.
- Imagem - Armazena um bit no LSB de cada pixel, em cada canal de cor. É a forma mais simples de fazer esteganografia em imagens.
- Composto - Agrega uma série de objetos de cobertura. Desta forma, é possível dispersar a mensagem em diversos arquivos de cobertura, por exemplo, em todas as imagens de uma pasta.
- Dummy - Qualquer que seja o conteúdo do arquivo, tem capacidade zero. Usado para inserir arquivos não suportados em um objeto de cobertura composto.

É importante notar que esta camada não foi o foco principal do projeto, e que existem técnicas muito mais sofisticadas do que as implementadas.

3.2. Algoritmo de Espalhamento

Dado o objeto de cobertura, que é visto como uma coleção de posição onde podemos armazenar os bits da mensagem, o algoritmo de espalhamento "sorteia" a posição onde o próximo bit será lido/armazenado. Em seguida, esta posição é removida da coleção, a fim de evitar sobreposição dos dados. É essencialmente o algorítmo Knuth-Shuffle sob-demanda.

Um bom PRNG deverá ser imprevisível. Desta forma, a mensagem, mesmo que detectada, não poderá ser recuperada pelo atacante, uma vez que praticamente qualquer mensagem com tamanho compatível poderá ser recuperada ao ordenar os bits disponíveis adequadamente. Por outro lado, a fim de que o remetente e o destinatário recebam a mesma permutação dos bits, e consequentemente, a mesma mensagem, o PRNG deve ser determinístico.

Para conciliar as características *imprevisível* e *determinístico*, é necessário inicializar adequadamente o PRNG com entropia antes do uso, caso contrário o atacante poderá prever o espalhamento e recuperar a mensagem. A terceira camada é responsável por gerar esta entropia de forma consistente entre remetente e destinatário, mas imprevisível ao atacante.

3.3. Gerador de Entropia

Caso a mensagem seja detectada, o gerador de entropia é responsável por torná-la indecifrável para o atacante. Para isso, obtém entropia das seguintes fontes principais:

- Senha - A forma mais simples consiste em uma chave compartilhada entre remetente e destinatário. Devido às dificuldades de se fazer a troca de chaves de maneira segura, esta fonte de entropia nem sempre é possível.
- Diffie-Hellman - Utilizando o protocolo Diffie-Hellman, é possível estabelecer uma chave compartilhada a partir da chave privada do remetente e da chave pública do destinatário e vice-versa. Desta forma, ambos conseguem obter a chave compartilhada, mas o atacante não a obtém. Os parâmetros P, G e as chaves públicas devem ser trocas separadamente, mas não precisam ser feitos de forma segura.
- Elliptic Curve Diffie–Hellman - Utilizando o protocolo Elliptic Curve Diffie–Hellman. É conceitualmente equivalente a Diffie-Hellman, porém utilizando criptografia de curvas elípticas. Os parâmetros da curva elíptica e as chaves públicas devem ser trocas separadamente, mas não precisam ser feitos de forma segura.

Porém o uso destas mesmas chaves repetidamente abre uma oportunidade de esteganoanálise. O uso de chaves efêmeras é em geral inviável em esteganografia, uma vez que o canal de comunicação muitas vezes é unidirecional, lento e não confiável.

Para mitigar este problema, algumas fontes de entropia adicionas são utilizadas implicitamente pela nossa solução. Estas fontes podem ser facilmente reproduzidas pelo atacante, mas suas iterações com as chaves anteriores na geração de entropia são extremamente difíceis de se atacar, de forma que as mesmas chaves poderão ser usadas repetidamente sem problemas de esteganoanálise.

- Objeto de Cobertura - Uma vez que o objeto de cobertura deveria ser descartável (Caso contrário, a detecção da mensagem fica trivial ao comparar duas mensagem sobre o mesmo objeto de cobertura), esta é uma boa

fonte de entropia. Para isso, calculamos um hash modificado do objeto, no qual os bits disponível para armazenamento são ignorados, de modo a manter o hash inalterado após a inserção da mensagem.

- Mensagem - Caso o mesmo objeto de cobertura e as mesmas chaves sejam utilizados, a permutação dos bits será constante. Para atenuar este problema, após a leitura / escrita de cada bit, este é fornecido ao PRNG como uma fonte adicional de entropia. O problema é que o espalhamento irá ser modificado apenas quando o primeiro bit diferente for encontrado, o que não será adequado caso a mensagem possua um cabeçalho invariante.

- Cabeçalho aleatória - A fim de amenizar o problema anterior de mensagens com cabeçalhos, um cabeçalho aleatório de alguns bits pode ser inserido antes da mensagem. Este cabeçalho deverá ser criado por outra fonte de números aleatórios, não determinística. Esta é a única técnica que ocupa parte da capacidade de armazenamento do objeto de cobertura, uma característica nao desejada e por isso opcional.

3.4. Estratégias de Criptografia Assimétrica

Ao se basear no protocolo Diffie-Hellman, fica implícito a existência de chaves do remetente e do destinatário.

Em alguns casos, esta característica pode ser problemática. Se o remetente não teve a chance de efetuar a troca de chave com o destinatário previamente, este não terá como saber a chave pública do remetente sem levantar suspeita. No momento, o método mais promissor para lidar com este problema é a troca de chave por meio de uma terceira parte que não se encontre sob observação.

Neste caso, uma forma das várias formas de fazê-lo é gerar um par de chaves pública/privada efêmero para si, inserir sua chave pública no objeto de cobertura, e só depois utilizar Diffie-Hellman para alimentar a entropia do PRNG antes de inserir o corpo da mensagem. De forma simétrica, o destinatário lê a chave pública do objeto de cobertura, aplica Diffie-Hellman e lê o corpo da mensagem. Apenas o destinatário será capaz de fazer a leitura.

Existe também o problema inverso, no qual qualquer um poderá receber a mensagem (Se souber da existência dela no objeto de cobertura), mas em que o remetente precisa ser certificado (Assinatura da mensagem). Embora não um problema quando as chaves já tenham sido trocadas previamente, o tratamento deste noralmente envolve uma autoridade certificadora.

Note que o uso de bits adicionais para transporte de chaves requer espaço adicional do objeto de cobertura. Com isto, diminuímos o tamanho máximo da mensagem, ao mesmo tempo que tornamos a detecção da mensagem escondida mais fácil. Em especial, as chaves utilizadas por Diffie-Hellman costumam ser bastante grandes, tipicamente 1024 bits. Por esta razão, preferimos a versão do algoritmo com curvas elípticas, na qual chaves com segurança equivalente ocupam cerca de 10X menos espaço. Sugerimos evitar esta estratégia quando possível.

3.5. Espalhamento Probabilístico

Até agora, nossa técnica não tinha se focado em dificultar a detecção da mensagem, mas sim em torná-la recuperável apenas pelo destinatário.

Nesta seção, partimos do princípio que nem todos os bits armazenados no objeto de cobertura tem a mesma probabilidade de detecção. Se usarmos como exemplo a inserção LSB em uma imagem, ferramentos de software podem perceber com facilidade um pixel alterado em uma região homogênea da imagem.

Como exemplo simples, podemos usar a ferramenta de preenchimento do Paint para colorir de preto uma região totalmente branca. Se houver inserção LSB, os pixels alterados não serão coloridos, tornando-se facilmente visíveis.

Em uma região não uniforme (Bordas, folhagem, etc), tais alteraçoes são muito mais difíceis de se detectar. Desta forma, a comunicação entre o objeto de cobertura e o algoritmo de espalhamento passa a dizer não apenas a quantidade de bits disponíveis, mas também a probabilidade (relativa) de cada um deles ser sorteado. Bits de difícil detecção receberão grandes probabilidades de uso, e bits facilmente detectáveis receberão probabilidades baixas (Ou mesmo nulas).

O algoritmo de espalhamento faz um sorteio probabilístico do bit escolhido. Bits de alta probabilidade vão (provavelmente) ser escolhidos primeiro. Caso a mensagem sendo armazenada seja pequena, apenas os bits de difícil detecção serão usados, tornando a mensagem muito difícil de detectar. Por outro lado, se a mensagem for muito grande, todos os bits acabarão sendo utilizados, portanto não há comprometimento da capacidade máximo do objeto. Desta forma, a técnica encontra balanço entre a capacidade de armazenamento e a dificuldade de detecção.

3.6. Implementaç^{a}o de Espalhamento Probabilístico

A implementação deste método mostrou-se mais complicada que a anterior: Ao considerar todas as probabilidades iguais é possível armazenar o índice corresponde a cada bit em um vetor. O sorteio consiste simplesmente em pegar o índice em uma posição aleatória do vetor, e em seguida remover este elemento do vetor (trocando-o com o último elemento e decrementando o tamanho). A implementação é simples e muito rápida.

Porém não é possível utilizá-la para sorteios probabilísticos. A estratégia que encontramos para fazer tais sorteios se baseia em estruturar os bits em uma árvore binária onde:

As folhas correspondem a cada bit disponível, e armazenam a probabilidade correspondente. Os demais nós armazenam a soma da probabilidade dos nós-filhos, e possui exatamente 2 filhos. O sorteio consiste em encontrar um número aleatório P no intervalo [0..somas das probabilidades-1]. O bit correspondente é encontrado da seguinte maneira recursiva:

- Se o nó for uma folha
  - Retorna o bit correspondente.
- Senão
  - Se P for menor que a soma das probabilidades no primeiro filho
    - Procura recursivamente no primeiro filho
  - Senão
    - Subtrai a soma das probabilidades no primeiro filho de P
    - Procura recursivamente no segundo filho

Após encontrar o bit correspondente, é necessário removê-lo da árvore para evitar sorteios com repetição.

Apesar deste algoritmo funcionar corretamente e ser facilmente classificado como O(N log(N)) (A árvore é inicialmente balanceada), notamos que a performance era muito ruim.

O gargalo da técnica estava na alocação dinâmica de grande quantidade de nós da árvore (2X a capacidade do objeto de cobertura).

Para resolver este problema de performance, atualmente usamos uma estratégia mista, na qual os bits de mesma probabilidade são agrupados. Dentro destes grupos, a primeira abordagem, na qual um elemento do vetor é sorteado e removida a cada vez, pode ser implementada eficientemente.

Em seguida, as árvores binárias do segundo método são utilizadas para unir os conjuntos de mesma probabilidade, de forma que os nós folha da árvore, ao invés de corresponder a um único bit, corresponde ao conjunto de todos os bits de mesma probabilidade.

3.7. Espalhamento Probabilístico em Imagens

A técnica descrita foi implementada apenas para imagens. Nos demais tipos de arquivos suportados, foi dada probabilidade 1 a todos os bits.

Para imagens, a probabilidade de cada bit é calculada pela variância dos valores dos pixels adjacentes em uma janela 3x3. Este cálculo é efeturado separadamente para cada canal de cor.

Note que, para região homogêneas da imagem, a variância será zero, ou seja, tal região torna-se inutilizável para armazenamento.

Figure 1. Espalhamento probabilístico 1, Imagem original e mapa de probabilidades

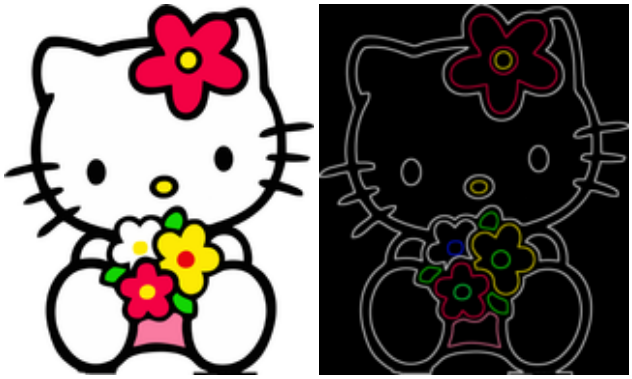

Figure 2. Espalhamento probabilístico 2, Imagem original e mapa de probabilidades

4. Experimentos e Discussão

Experimento 1 - Foto Natural de Praia: note que, se necessário, a imagem toda pode ser utilizada para armazenar a mensagem. Caso contrário, as região de detecção mais difíceis (Folhagens e bordas) são ocupadas primeiro, seguidas de regiões intermediárias (Corais, nuvens, etc), e por último são utilizadas as regiões praticamente homogêneas do céu, mar e areia.

Exemplo 2 - Clipart da Hello Kitty: imagem original e mapa de probabilidades, note que neste tipo de imagem, apenas as bordas são utilizadas, comprometendo a capacidade de armazenamento.

5. Conclusões e Trabalhos Futuros

O programa foi capaz de mapear usando uma chave assimétrica como fator de entropia para tornar a mensagem a ser escondida indecifrável sem uso direto de criptografia, e foi possível incorporar outras técnicas para dificuldar a identificação da presença da mensagem. Linhas promissoras de estudo incluem o uso de outras técnicas de esteganografia já disponíveis para aprimorar a segurança do método e o estudo de formas para garantir a robustez do mapeamento em casos nos quais o arquivo sofra modificações, como compressão com perdas ou redução da qualidade da imagem.

6. Referências

[1] NIU Xiam; ZHOU C.; DING Jianghua; YANG Bian; JPEG Encryption with File Size Preservation, Intelligent Information Hiding and Multimedia Signal Processing, 2008. IIHMSP '08 International Conference on

[2] BELLARE M., RISTENPART T., ROGAWAY P., STEGERS T.; Format-Preserving Encryption

[3] ISHIDA Takayuki, YAMAWAKI Kazumi, NODA Hideki, NIIMI Michiharu; An Improved QIM-JPEG2000 Steganoghaphy and Its Evaluation by Steganalysis, Journal of Information Processing Vol. 17 267-272 (Oct. 2009)